# Détection de pose de véhicule pour la reconnaissance de marque et modèle.

Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Thierry Chateau, Céline Teulière

# Détection de pose de véhicule pour la reconnaissance de marque et modèle

# Pose Detection for vehicle Make and Model Recognition

F.Chabot [1] M.Chaouch [1] C.Teulière [2] J.Rabarisoa[1] T.Chateau[2]

[1]CEA, LIST, Laboratoire Vision et Ingénierie des Contenus, France
[2]Institut Pascal, UMR6602, CNRS, Université Blaise Pascal, Clermont-Ferrand, France

F-91191 Gif-sur-Yvette, France
florian.chabot@cea.fr

## Résumé

*Nous présentons une nouvelle méthode de détection de pose d'un véhicule dans une image dans le but de reconnaître sa marque et son modèle. Notre approche repose sur la mise en correspondance entre le véhicule dans l'image et des modèles 3D rigides. En utilisant un détecteur fondé sur des réseaux de neurones convolutionnels (CNN), des points d'intérêts correspondant à des parties prédéfinies sur le véhicule sont extraits sur l'image. Ces points seront ensuite filtrés et mis en correspondance avec les points des modèles 3D. Notre méthode permet d'améliorer les performances de la détection de pose et est plus adaptée pour la reconnaissance de marque et modèle de véhicules que des approches basées sur un modèle déformable.*

## Mots Clef

Détection de pose, Mise en correspondance, Points d'intérêts, Réseaux de Neurones Convolutionnels

## Abstract

*We propose a new approach for vehicle pose estimation with the goal of make and model recognition. Our algorithm is based on the matching between the vehicle in the image and 3D models. With a detector based on Convolutionnal Neural Networks (CNN), interest points corresponding to predifined parts are extracted in the image. These points are then filtered and matched with 3D models points. Compared to approaches based on active shape models, our approach increases pose estimation accuracy and improves make and model recognition.*

## Keywords

Pose estimation, Matching, Interest points, Convolutional Neural Networks

## 1 Introduction

La reconnaissance de marques et de modèles de véhicules permet des applications diverses dans les systèmes de transport intelligent et de surveillance automatique telle que le suivi, le recensement, le contrôle de véhicules, ou encore l'automatisation de l'accès. Ce type de reconnaissance est un défi dans le domaine de la vision par ordinateur. En effet, discriminer des objets parfois très semblables, comme par exemple deux « berline tricorps » est très complexe. Le problème traité ici est au-delà d'un problème d'une reconnaissance de classes d'objets sémantiquement et visuellement différents ( personne, voiture, bus, moto, avion…). Notre objectif final est de reconnaître des sous-classes de l'objet « véhicule » très proches visuellement. Nous parlons d'une reconnaissance fine de d'objets.

La grande majorité des méthodes de reconnaissance de marque et de modèle de véhicules [1, 2] nécessite une détection de pose préalable. Plus celle-ci est précise, plus les contraintes d'apparence liées au point de vue disparaissent et permettent d'orienter les algorithmes de classification vers de la reconnaissance fine.
Nous proposons ici un détecteur de pose basé sur l'appariement de modèles 3D avec le véhicule dans l'image. Des points d'intérêts correspondant à des parties du véhicule sont extraits à l'aide d'un détecteur multi-classes appris avec des réseaux de neurones convolutionnels (CNN) sur une base de données synthétique de grande taille. Deux phases de post-traitement sont intégrées dans notre approche pour regrouper et sélectionner les détections présentant une cohérence spatiale. Le fait d'utiliser la géométrie entre parties permet une mise en correspondance robuste entre le véhicule dans l'image et son modèle 3D.
La Fig.1 donne une vision d'ensemble de l'approche proposée qui sera par la suite utilisée comme première étape d'un algorithme de reconnaissance de marque et modèle de véhicules.

Notre choix s'est porté sur une méthode d'apprentissage automatique basée sur les réseaux de neurones convolutionnels largement abordés dans la littérature de ces dernières années. En effet, les CNN ont démontré leur efficacité dans les problèmes de classification tels que la détection d'objet [3, 4], la segmentation [3] et la reconnaissance de visage [5].
Nous présentons dans la section suivante l'état de l'art. Nous détaillons dans la troisième section la méthode proposée pour l'estimation de la pose. Nous présentons les résultats et l'étude comparative à une méthode de l'état de l'art dans la section 4.

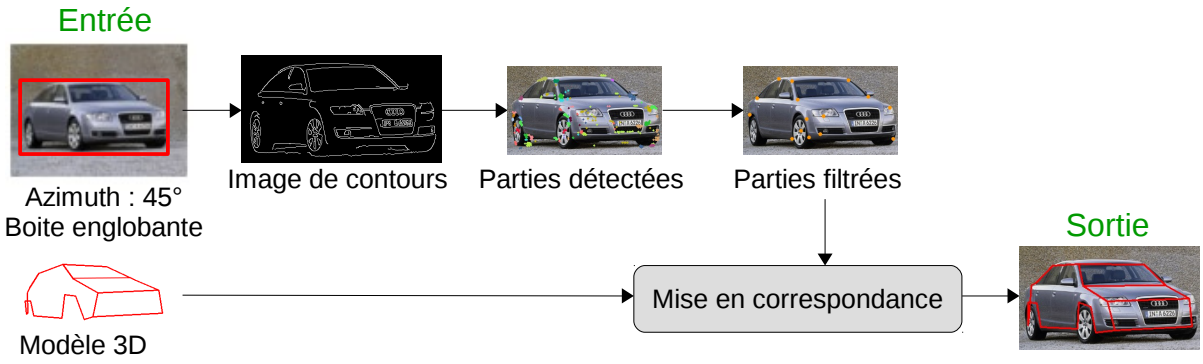


*Fig. 1 : Vue d'ensemble du système proposé. L'algorithme consiste à détecter les contours dans l'image puis procéder à une détection de parties. Ces détections sont ensuite filtrées et mise en correspondance avec un modèle 3D.*

## 2 Etat de l'art

Notre étude bibliographique s'est concentrée sur deux problématiques. La première est la reconnaissance fine d'objet, en particulier la marque et le modèle de véhicules. La seconde est la détection de pose qu'elle soit grossière ou fine.

Pour la reconnaissance de marque et modèle de véhicules, Medioni et al. [1] introduisent une mesure de similarité basée sur la mise en correspondance entre points d'intérêts SIFT détectés sur des modèles 3D texturés et sur l'image d'entrée. La méthode proposée dans [2] repose sur des modèles de courbes 3D propres à chaque modèle de véhicules et effectue une mise en correspondance basée contours afin de détecter la marque et le modèle. Les auteurs de [6][7] s'intéressent à la reconnaissance fine d'objets et en particulier la classe « chaise » et non « véhicule ». Ils apprennent un grand nombre de détecteurs multi-vues DPM (Deformable Part Models) [8] sur des rendus de modèles 3D de chaises.

Pour la détection de pose grossière, les méthodes [9, 10, 11, 12] proposent de discrétiser les points de vue en 8 ou 16 azimuths. S'inspirant du DPM, les travaux [9, 10] proposent le VDPM (Viewpoint-DPM) où chaque classe à apprendre correspond à un point de vue discret. Liebelt et Schmid [11] proposent une méthode de détection d'objets multi-vues en utilisant un modèle d'apparance de partie appris sur des images réelles et un modèle de cohérence géométrique 3D construit à partir de modèles synthétiques 3D.

Plusieurs méthodes de détection de pose précise [13, 14, 15, 16] utilisent les méthodes de détection de pose grossière comme initialisation. Les auteurs de [17] partent du principe que l'objet est connu au préalable et mettent en correspondance son modèle 3D avec l'image par minimisation d'une fonction de coût combinant plusieurs caractéristiques géométriques et d'apparence. D'autres méthodes utilisent un a priori faible de l'objet dans l'image, par exemple la classe « véhicule ». Ainsi, les auteurs de [16] proposent d'utiliser un modèle 3D moyen de parties pour les mettre en correspondance avec celles de l'image tout en respectant une cohérence spatiale. Pepik et al. [18] proposent une nouvelle méthode appelée DPM 3D en s'inspirant du DPM [8] pour détecter l'objet et son point de vue. Zia et al. [13, 14, 15] introduisent dans leur approche un modèle déformable 3D (*ASM Active Shape Model* [19]). En faisant varier les paramètres de forme et de caméra, plusieurs configurations sont générées et projetées dans l'image. La configuration maximisant le score (provenant d'un détecteur de parties) sur l'ensemble des sommets projetés est retenue.

## 3 La méthode d'estimation de la pose

Nous présentons dans cette section notre méthode de détection de pose précise de véhicules appelée *Deep Pose Car* (DeepPC).
Les deux entrées de notre algorithme sont la boîte englobant le véhicule et la pose grossière qui correspond à la discrétisation de l'ensemble des azimuths comme le montre la Fig. 1. Plusieurs méthodes [8 ,9, 10] permettent de calculer ces deux entrées (boîte, pose grossière) automatiquement. Dans ce travail, nous ne nous intéressons pas à cette étape d'initialisation.

Notre méthode de détection de pose précise se décompose en deux étapes : la détection des parties et la mise en correspondance. La détection des parties (points d'intérêts) est effectuée avec un détecteur multi-classes appris avec des réseaux de neurones convolutionnels. Les hypothèses résultant de cette détection sont groupées et filtrées en prenant en compte les contraintes géométriques propres aux véhicules. La mise en correspondance permet d'apparier les parties détectées avec les points correspondants dans les modèles 3D.

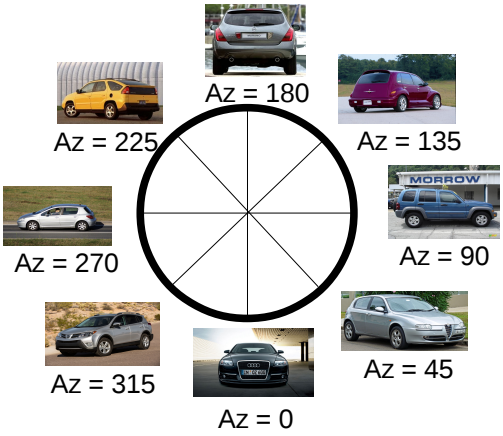


*Fig. 2 : Représentation des 8 azimuths grossiers supposés connus.*

## 3.2 Détection des parties

La détection de parties qui seront mises en correspondance se décompose en trois phases. La première phase consiste à extraire une image de contours à partir de l'image d'entrée et à détecter sur celle-ci les parties en utilisant un modèle appris avec des CNN sur des données synthétiques. La seconde phase permet de regrouper les hypothèses proches associées à la même partie. La dernière phase est un filtrage par cohérence spatiale permettant de garder une seule détection par partie.

### 3.2.1 Entrainement des CNN

Pour générer un nombre conséquent d'exemples positifs sans procéder à un étiquetage important sur une base de données d'images de véhicules réels, nous avons choisi d'extraire des patchs synthétiques par projection de modèles 3D. Ces exemples sont utilisés pour entrainer les CNN. Les sommets des modèles 3D correspondants aux points d'intérêts que l'on cherche à détecter sont annotés (36 points 3D annotés sur chaque modèle). Pour extraire des caractéristiques communes entre les images synthétiques (données d'apprentissage) et les images réelles (données à tester) nous avons opté pour l'extraction des contours saillants sur les modèles 3D et les contours sur les images 2D avec des méthodes basées Canny [20, 21]. En effet, les contours saillants 3D correspondent aux lignes connectant deux faces dont les normales forment un angle suffisamment grand (plus de 60° dans notre cas). Ces lignes sont localisées aux endroits où la variation d'intensité pixellique semble la plus forte dans une image. Des travaux ont déjà utilisé ces rendus « contours » comme descripteurs pour la compréhension de scène [14, 15, 22]. Ainsi, un nombre très important de données (nécessaire pour l'apprentissage des CNN) ont été générées en projetant les modèles 3D suivant différentes matrices de projection. De plus, bien que les CNN ont pour but de ne pas avoir à calculer des caractéristiques sur les données avant d'effectuer l'apprentissage, le fait de les apprendre sur des données synthétiques nous oblige a appliquer une première couche de calcul de caractéristiques (calcul des contours) pour introduire une cohérence entre les données apprises et les données à tester mais permet de s'affranchir d'un effort de labellisation important.

Pour procéder à la détection des parties un CNN est appris pour chaque azimuth grossier. La sortie de chaque réseau consiste en $N+1$ classes avec $N$ le nombre de parties visibles dans l'azimuth associé au réseau plus une classe de « background ». Chaque réseau possède deux étapes d'extraction de caractéristiques. Chacune de ses étapes est caractérisée par une couche de convolution, une fonction d'activation (tangente hyperbolique) et une couche de pooling (max pooling). Les filtres de convolution appris sont de taille 5 x 5. La partie du CNN dédiée à la détection (réseaux de neurones complètement connectés) est constituée d'une couche cachée et d'une couche de sortie à $N+1$ classes. La Fig. 3 présente l'architecture des CNN utilisés.

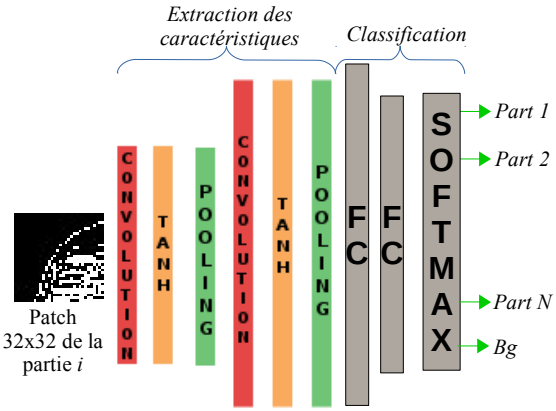


*Fig. 3 : Architecture des CNN appris sur des patchs extraits de données synthétiques.*

### 3.2.2 Détection des parties avec les CNN

Une carte de contours de l'image est tout d'abord calculée grâce à l'algorithme proposé dans [21]. Nous effectuons ensuite une détection par fenêtre glissante en utilisant le détecteur correspondant à l'azimuth grossier de l'image. La Fig. 4 montre les résultats de la détection de parties avec un réseau de neurones appris sur l'azimuth 45.

*Fig. 4 : Première ligne, l'image à tester et sa carte de contours sur laquelle va être effectuée la détection. Deuxième ligne, les parties détectées. Chaque couleur correspond à une partie.*

L'implémentation de la fenêtre glissante est inspirée de [23]. Au lieu d'extraire des patchs 32 x 32 sur la carte de contours et de les faire passer un par un dans le réseau complet pour prédire sa classe, nous avons divisé la détection en deux étapes : l'extraction des caractéristiques et la classification. L'extraction des caractéristiques se fait sur l'ensemble de l'image, ce qui permet de calculer la suite des convolutions en une seule fois (cf Fig. 5).
À la fin de cette étape, une carte de caractéristiques est générée. C'est sur celle-ci que les patchs à classifier sont extraits (Fig. 6). Contrairement à [14], chaque patch est attribué à une classe qui correspond à celle qui a le score maximal. La suite de l'algorithme n'utilisera pas de carte de score mais une carte de classes.
La détection de parties fournit un ensemble d'hypothèses de points d'intérêts auxquelles sont associés des points 3D dans la base de modèles 3D. Notre motivation étant de garder une seule détection par partie, nous cherchons à réduire les bonnes détections redondantes et supprimer le maximum de fausses détections.

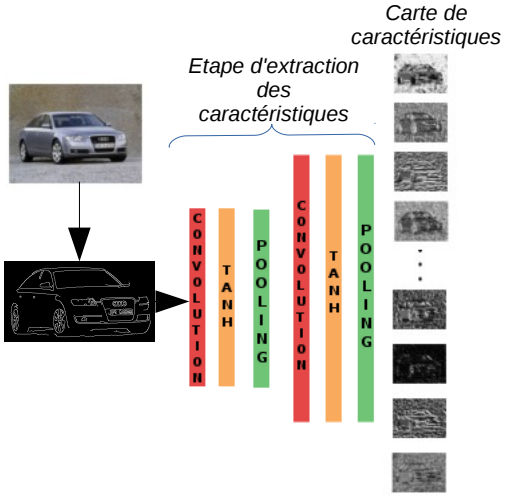


*Fig. 5 : Étape d'extraction des caractéristiques*

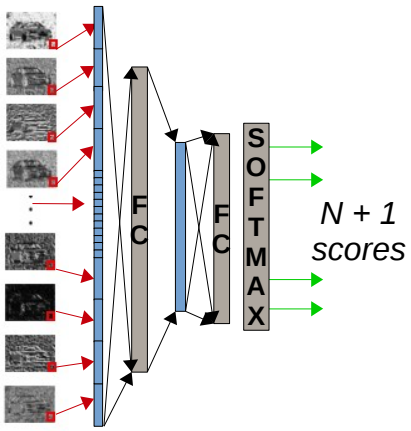


*Fig. 6 : Classification par extraction de patchs sur la carte de caractéristiques (fenêtre glissante). Les briques grises correspondent aux couches du CNN complètement connectées (fully connected).*

### 3.2.3 Agglomération des détections

Nous proposons d'appliquer un clustering sur les détections résultantes de la phase précédente, l'objectif étant de réduire leur nombre. Il s'agit de regrouper les détections associées à une même partie et agglomérées dans un voisinage proche. L'algorithme *k-means* [24] est appliqué sur l'ensemble des hypothèses de chaque partie. La Fig. 7 illustre le résultat de ce clustering.
À chaque partie $p$ sera associée un nombre $k_p$ de clusters. L'idée est de trouver la valeur de $k_p$ maximum tel que l'ensemble des distances entre les centres des clusters soient supérieures à un *seuil*. L'estimation du nombre de clusters correspondant à l'ensemble des hypothèses d'une partie est calculée comme suit :

$$k_p = max_k\left(D_p(k), D_p(k) = \frac{k(k+1)}{2}\right) \quad \text{avec}$$

$$D_p(k) = Card\{(i,j), \|c_i^p - c_j^p\| > seuil, 1 \leqslant i < j \leqslant k\}$$

$c_i^p$ correspond au centre du cluster $i$ des détections associées à la partie $p$, le *seuil* est fixé pour une image normalisée par rapport à la hauteur de la boîte englobant le véhicule (pour nos expériences le *seuil* est fixé à 10 pixels pour une hauteur de 155 pixels).

*Fig. 7 : Première ligne, les parties détectées. Deuxième ligne, les centres de clusters trouvés.*

### 3.2.4 Filtrage par cohérence spatiale

Après l'agglomération des détections, un filtrage par cohérence spatiale est appliqué pour sélectionner une seule détection par partie détectée.
Pour cela, nous avons généré pour chaque modèle 3D un ensemble de configurations $C$ (azimuth, élévation, focale) que nous projetons dans la boîte englobant le véhicule. Environ 400k configurations ont été précalculées pour

chaque modèle et chaque azimuth grossier. L'idée ici est de projeter le modèle dans un maximum de configurations. Nous cherchons, pour chaque configuration, les hypothèses de parties qui se trouvent dans un voisinage des sommets projetés. Le filtrage consiste à retenir la configuration qui maximise le nombre d'hypothèses dans le voisinage des sommets projetés et qui minimise la distance entre ces sommets et les hypothèses des parties. La Fig. 8 illustre le fonctionnement du filtrage par cohérence spatiale,

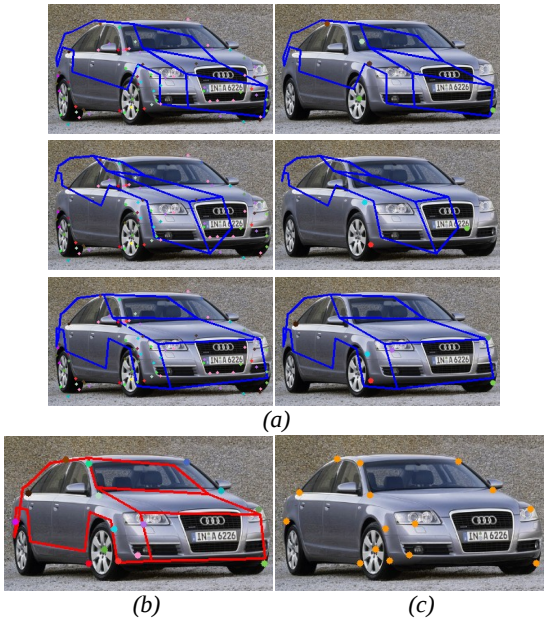

*Fig. 8 : Filtrage par cohérence spatiale. (a) 3 exemples de configurations projetés sur le véhicule (colonne de gauche) et les parties filtrées qu'elles engendrent (colonne de droite). (b) La configuration retenue en rouge. (c) les parties filtrées.*

Ainsi pour chaque configuration $c \in C$, on calcule :

$$d_c^j = min_i(\|q_c^j - p_i^j\|) ,$$

$$n_c = Card(\{d_j^c\}, d_j^c < seuil, j \in \{1, N\}) ,$$

$$d_c = \sum_{j, d_j^c < seuil} d_j^c , \quad D_c = \frac{d_c}{n_c} ,$$

$q_c^j$ étant le point projeté dans la configuration $c$ associé au point 3D de la partie $j$ annotée dans le modèle 3D et $p_i^j$ étant le point correspondant à l'hypothèse $i$ de la partie $j$. Finalement, on retiendra la configuration $C_r$ telle que :

$$C_r = argmin_c\{D_c, n\} ,$$

avec $$n = argmax_c\{n_c\} .$$

## 3.3 Mise en correspondance

L'étape du filtrage par cohérence spatiale permet de détecter des points d'intérêts proches de la solution optimale. Le but de la mise en correspondance est de retrouver la pose précise en partant de cette solution.
Nous utilisons alors une méthode de détection de pose 2D/3D [25] itérative basée sur l'optimisation de Levenberg-Marquardt pour faire la mise en correspondance de ces points 2D filtrés et les points des modèles 3D (Fig. 9). Ce calcul de pose est robuste aux mauvaises détections (outliers) de par l'utilisation d'un tirage aléatoire de type RANSAC.

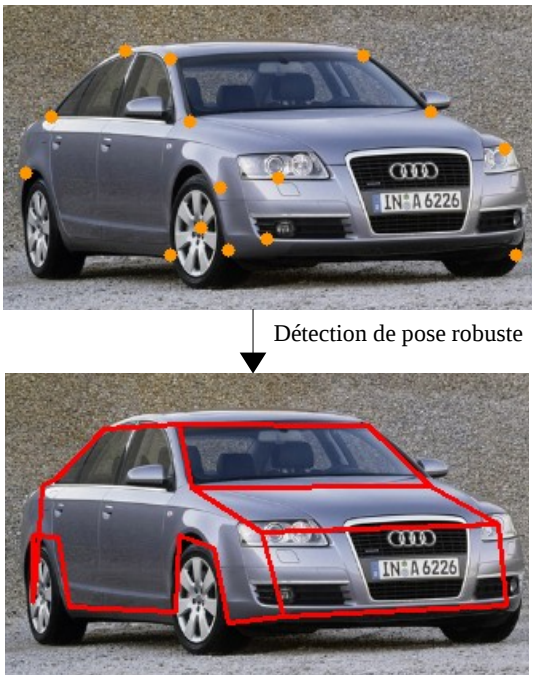


*Fig. 9 : Calcul de la pose par la mise en correspondance du modèle 3D du véhicule avec l'image.*

## 4 Expérimentations

Cette section présente les expérimentations effectuées pour évaluer notre méthode par rapport à l'état de l'art.
Notre base de données de modèles 3D est constituée de 36 modèles de voitures. Plusieurs expérimentations ont été réalisées afin d'une part, d'évaluer la capacité de l'algorithme à mettre en correspondance ces modèles sur l'image, et d'autre part, prédire la marque et le modèle du véhicule en sélectionnant le modèle 3D le mieux mis en correspondance. La Fig. 13 montre certains résultats obtenus avec l'algorithme présenté.

**Génération des données d'apprentissage**

Comme [14] les données d'apprentissage positives sont des patchs 32 x 32 centrés sur les sommets annotés extraits de rendus « contours » des modèles 3D. Ces données ont été bruitées afin de les rendre plus réalistes: des morceaux de ligne et d'ellipse ont été ajoutés

aléatoirement dans les données ainsi qu'un bruit impulsionnel. Les négatifs sont, d'une part, des patchs 32 x 32 extraits aléatoirement de PASCAL VOC 2007 auxquels a été appliqué un détecteur de contours de type Canny et d'autre part, des patches 32 x 32 extraits des rendus 3D qui ne correspondent pas aux parties (cf Fig. 10). Nous avons généré les exemples positifs en faisant varier les paramètres intrinsèques et extrinsèques de la caméra pour tous les modèles 3D de la base (36 modèles). Cette manière de procéder permet de générer un grand nombre de données en évitant un effort d'étiquetage trop important. Chaque CNN apprend sur environ 1,5 millions de patchs (500 000 de positifs et 1 million de négatifs) en environ 12h.

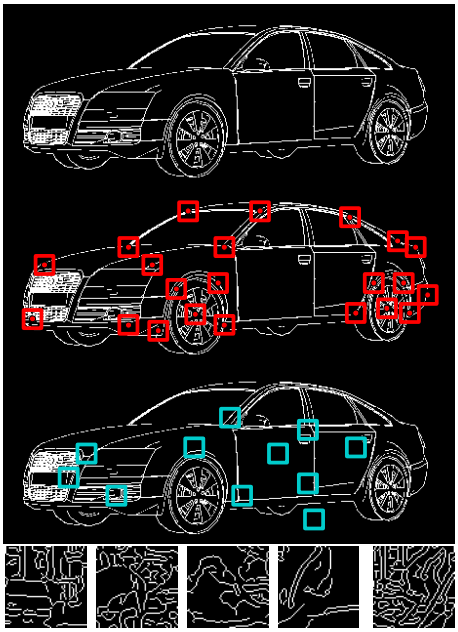

*Fig. 10 : Première ligne, rendu « contours » dans une certaine configuration (azimuth, élevation, focale). Deuxième ligne, les patchs utilisés comme positifs. Chaque patch correspond à une classe à apprendre. Troisième ligne : patchs de négatifs correspondant à la classe background. Dernière ligne : patchs de négatifs provenant de PASCAL VOC correspondant également à la classe background*

**Évaluation des détecteurs**

Nous avons dans un premier temps évalué les 8 détecteurs de parties. Nous avons pour cela généré des patchs de test provenant de rendus «contours» des modèles 3D. Ces données ont l'avantage d'être de la même nature (contours saillants) que les données d'apprentissage. Le Tableau. 1 regroupe les performances de détection pour chaque azimuth grossier sur cette base de test constituée d'environ 25 000 exemples de parties par azimuth.

| Azimuth | 0 | 45 | 90 | 135 | 180 | 225 | 270 | 315 |
|---|---|---|---|---|---|---|---|---|
| Performance (%) | 95,1 | 95,4 | 96,1 | 93,6 | 92,5 | 94,2 | 96,3 | 94,9 |

*Tableau. 1 : Perfomances des 8 CNN pour la détection des parties. La performance correspond au nombre de parties correctement classifiées sur le nombre total de patchs.*

**La base de données d'images réelles**

Il n'existe pas à ce jour de base de données mettant en relation des modèles 3D et des images réelles correspondant exactement à ces modèles. Nous avons donc constitué une base de données d'images provenant d'internet correspondant aux modèles 3D présents dans notre base. Cette base de données réelle est constituée de 400 images triées par azimuth grossier et par modèles. Les boîtes englobantes ainsi que les parties visibles dans chaque azimuth sont annotées.

**Protocole d'évaluation de la mise en correspondance**

Le protocole d'évaluation pour la mise en correspondance du modèle 3D avec le véhicule dans l'image est le même que [14]. Chaque image est normalisée par rapport à la hauteur de la boîte englobant le véhicule. Une partie est supposée bien mise en correspondance si sa distance à la partie de la vérité terrain est inférieure à un certain seuil (20 pixels). Pour ce travail, nous nous sommes comparés à [14] en nous affranchissant de leur modèle 3D déformable. Pour notre méthode et celle de [14], nous donnons le modèle 3D correspondant à l'image et nous évaluons la capacité des deux algorithmes à mettre en correspondance ce modèle 3D avec le véhicule. La Fig. 11 présente les résultats obtenus.

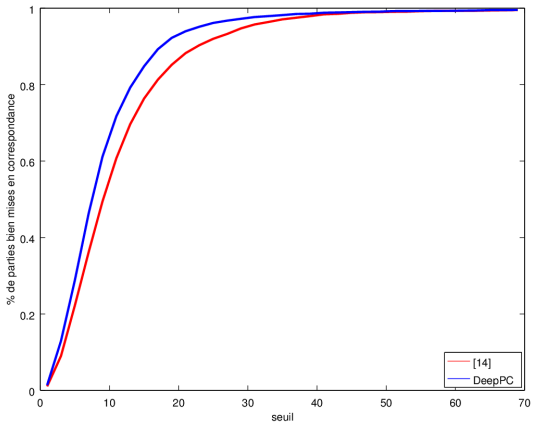


*Fig. 11 : Résultats de l'évaluation de la mise en correspondance (en rouge la courbe correspondant à [14], en bleu la courbe de notre méthode). En ordonnée : le pourcentage de parties bien mises en correspondance, en abscisse : la valeur du seuil permettant de considérer si une partie est bien appariée ou non.*

Nous pouvons remarquer que notre méthode de mise en correspondance est plus efficace que celle de [14] : pour un seuil de 20 pixels nous obtenons 93,2 % de parties bien mises en correspondance contre 86,9 % pour [14].

**Reconnaissance de marques et modèles de véhicules**
Pour finir, nous avons également évalué la capacité de l'algorithme à reconnaître la marque et le modèle du véhicule dans l'image d'entrée. Chaque modèle 3D est mis en correspondance sur l'image grâce à notre algorithme de détection de pose. Pour chaque modèle nous calculons l'erreur de reprojection et le nombre de points utilisés pour la détection de pose 2D/3D (inliers). Nous classons alors les modèles en nous basant sur ces deux critères : par ordre croissant suivant le nombre d'inliers puis par ordre décroissant suivant l'erreur de reprojection.
Sur ce point, nous nous sommes comparés à [15] qui propose une méthode de reconnaissance fine basée sur son modèle déformable 3D : l'ASM 3D mis en correspondance est comparé (plus proches voisins) à la base de modèles 3D afin d'être classifié. La Fig. 12 montre les résultats obtenus.

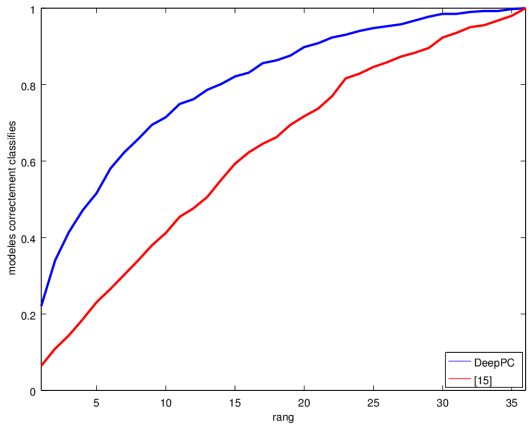


*Fig. 12 : Résultats de la reconnaissance de marque et modèle de véhicules. Les courbes représentent le pourcentage de marques et modèles trouvé pour un rang donné (notre méthode en bleu, celle de [15] en rouge)*

Notre mise en correspondance semble plus efficace que [15] lorsqu'il s'agit de détecter la marque et le modèle d'un véhicule : 22 % des modèles sont correctement trouvés au rang 1 contre 6,4 % pour [15]. Cependant, les critères utilisés pour classifier les modèles sont assez faibles et ne permettent pas de dissocier des classes de véhicules très semblables.

## 4 Conclusion

Pour conclure, la méthode de détection de pose proposée repose sur des détecteurs de points d'intérêts appris sur des données synthétiques. Les hypothèses de parties sont ensuite filtrées puis mises en correspondance avec les points de modèles 3D. L'objectif final de ce travail est la reconnaissance de marque et modèle de véhicules. Les résultats obtenus montrent que notre méthode est très efficace pour mettre en correspondance un modèle 3D avec le véhicule dans l'image. Pour reconnaitre la marque et le modèle du véhicule le critère utilisé semble trop peu discriminant même si nous obtenons de meilleurs résultats qu'une méthode de l'état de l'art. Pour la suite, nous prévoyons d'améliorer la détection de pose et la reconnaissance fine en raffinant l'algorithme existant avec une mise en correspondance plutôt basée sur les contours.

## Bibliographie


[1] J. Prokaj, G. Medioni, *3-D Model Based Vehicle Recognition*, WACV, 2009.

[2] K. Ramnah, S.N. Sinha, R. Szeliski, *Car Make and Model Recognition using 3D Curve Alignment*, WACV, 2009.

[3] R.B. Girshick, J. Donahue, T. Darrell, J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, arxiv:1311.2524, 2013.

[4] M. Oquab, L. Bottou, I. Laptev, J. Sivic, *Learning and transferring mid-level image representations using convolutional neural networks*, Technical Report HAL-00911179, INRIA, 2013.

[5] Y. Taigman, M. Yang, M.A. Ranzato, L. Wolf, *DeepFace : Closing the Gap to Human-Level Performance in Face Verificatio*n, CVPR, 2014,

[6] M. Aubry, D. Maturana, A. Efros, B. Russell, J. Sivic *Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models,* CVPR, 2014.

[7] J.J.Lim, A.Khosla, A.Torralba, *FPM : Fine pose Parts-based Model with 3D CAD models,* ECCV, 2014

[8] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D.Ramanan, *Object detection with discriminatively trained part based models*, PAMI, 2010.

[9] R.J. Lopez-Sastre, T. Tuytelaars, S. Savarase, *Deformable Part Models Revisited: A Performance Evaluation for Object Category Pose Estimation*, ICCV, 2011.

[10] Y. Xiang, R. Mottaghi , S. Savarese, *Beyond PASCAL : A Benchmark for 3D Object Detection in the wild*, WACV, 2014.

[11] J. Liebelt, C. Schmid, *Multi-View Object class Detection with a 3D Geométrique Model*, CVPR, 2010.

[12] M. Özuysal, V .Leptit, P. Fua, *Pose Estimation for Category Specific Multiview Object Localization*, CVPR, 2009.

[13] M. Zeeshan Zia, M. Stark, B. Schiele, K. Schindler, *Revisiting 3D Geometric Models for Accurate Object Shape and Pose*, PAMI, 2013.

[14] M. Zeeshan Zia, M. Stark, K. Schindler, *Explicit Occlusion Modeling for 3D Object Class Representations*, CVPR, 2013.

[15] M. Zeeshan Zia, M. Stark, B. Schiele, K.Schindler, *Detailed 3D Representations for Object Modeling and Recognition*, PAMI, 2013.

[16] Y. Xiang, S. Savareses, *Estimating the Aspect Layout of Object Categories*, CVPR, 2012.

[17] J.J. Lim, H. Pirsiavash, A. Torralba, *Parsing IKEA Objects: Fine Pose estimation*, ICCV, 2013.
[18] B. Pepik, M. Stark, P. Gehler, B. Schielen, *Teaching 3D Geometry to Deformable Part Models*, CVPR, 2012.
[19] T.F. Cootes, C.J. Taylor, D.H Cooper, J. Graham, *Active Shape Models – Their training and application*, Computer Vision and Image Understanding, 1995.
[20] J. Canny, *A Computational Approach To Edge Detection*, IEEE Trans. PAMI,1986.
[21] L. Wang, S. You, U. Neumann, *Supporting Range and Segment-based hysteresis thresholdong in edge detection*, ICIP, 2008.
[22] S. Satkin, J.Lin M.Herbert, *Data-Driven Scene Understanding from 3D Models*, BMVC, 2012.
[23] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, K. Keutzer, *DenseNet : Implementing Efficient ConvNet Descriptor Pyramids*, arxiv:1404.1869, 2014.
[24] J. MacQueen, *Some methods for classification and analysis of multivariate observations*. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, 281--297, University of California Press, Berkeley, Calif., 1967.
[25] Z. Zhang, *A Flexible New Technique for Camera Calibration,* PAMI, 2000

*Fig. 13 : Résultats de la détection de pose. A gauche, la détection de pose lorsque le modèle à mettre en correspondance est connu. A droite, la détection de pose lorsque le modèle n'est pas connu. Le modèle affiché est celui qui minimise l'erreur de reprojection sur l'ensemble des modèles et qui maximise le nombre d'inliers.*